Lab Exercise 6– Terraform Multiple tfvars Files

Objective:

Learn how to use multiple tfvars files in Terraform for different environments.

Prerequisites:

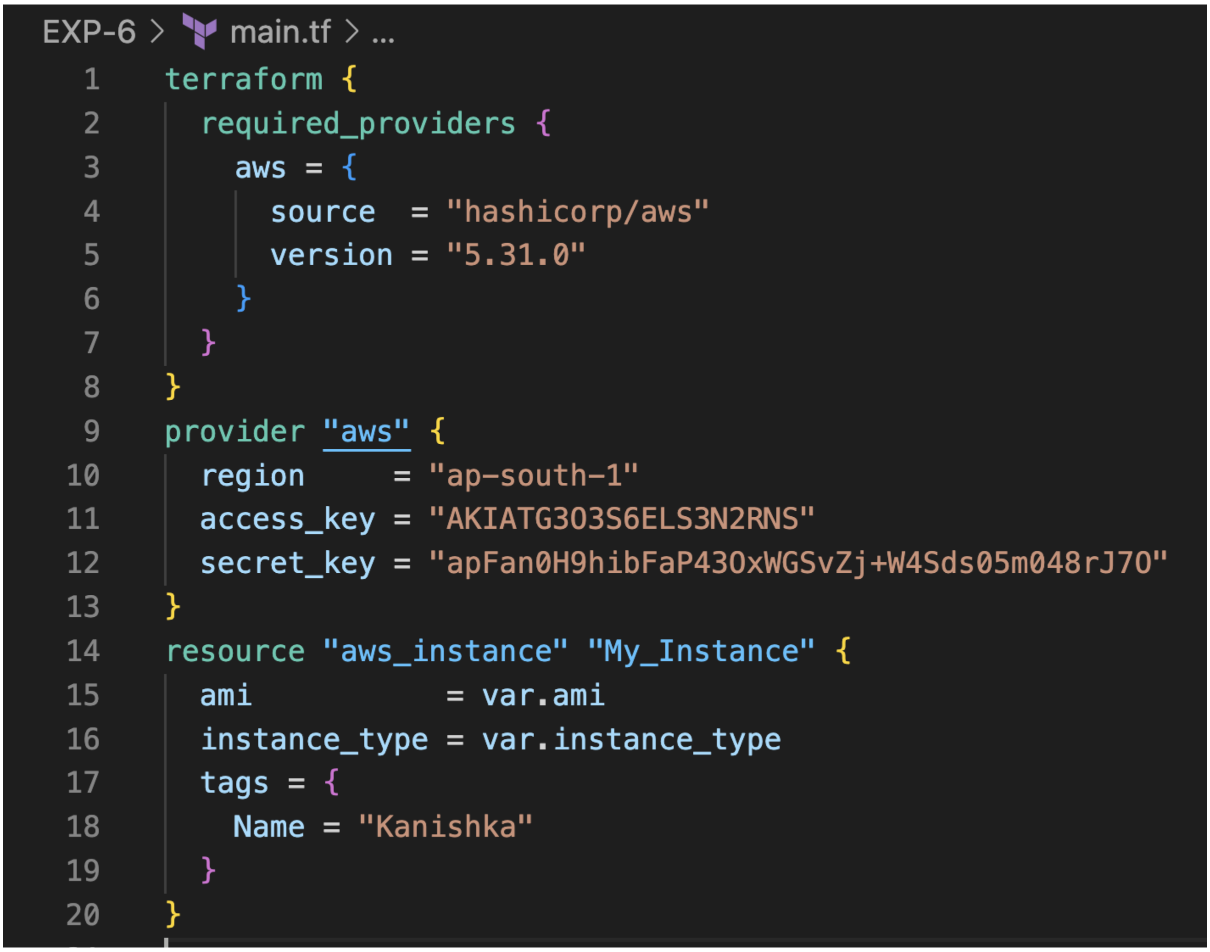
- Terraform installed on your machine.
- Basic knowledge of Terraform configuration and variables.

Steps:

1. Create Terraform Configuration Files:

- Create a file named main.tf:

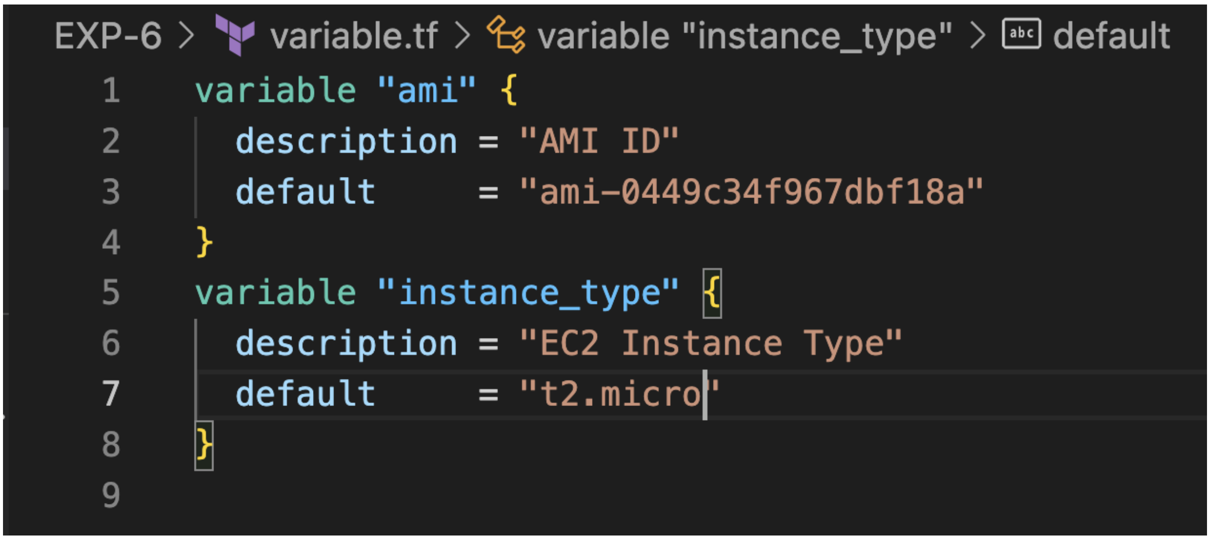
# main.tf



```
terraform {
  required_providers {
    aws = {
      source  = "hashicorp/aws"
      version = "5.31.0"
    }
  }
}
provider "aws" {
  region     = "ap-south-1"
  access_key = "AKIATG3O3S6ELS3N2RNS"
  secret_key = "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"
}
resource "aws_instance" "My_Instance" {
  ami           = var.ami
  instance_type = var.instance_type
  tags = {
    Name = "Kanishka"
  }
}
```

- Create a file named variables.tf:

# variables.tf

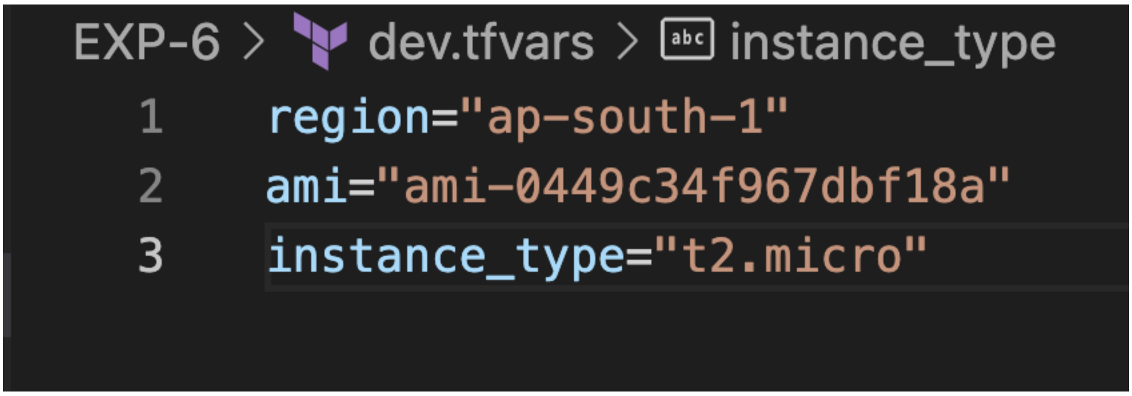


```
variable "ami" {
  description = "AMI ID"
  default     = "ami-0449c34f967dbf18a"
}
variable "instance_type" {
  description = "EC2 Instance Type"
  default     = "t2.micro"
}

```

2. Create Multiple tfvars Files:

- Create a file named dev.tfvars:

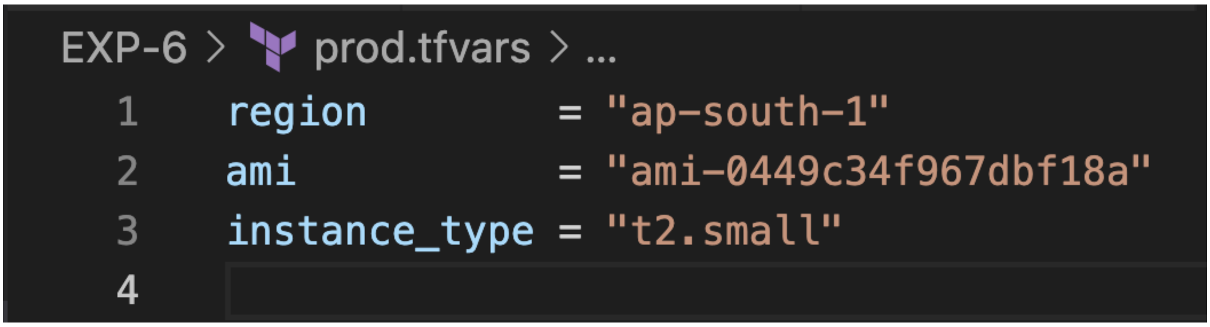
# dev.tfvars



```
region="ap-south-1"
ami="ami-0449c34f967dbf18a"
instance_type="t2.micro"
```

- Create a file named prod.tfvars:

# prod.tfvars

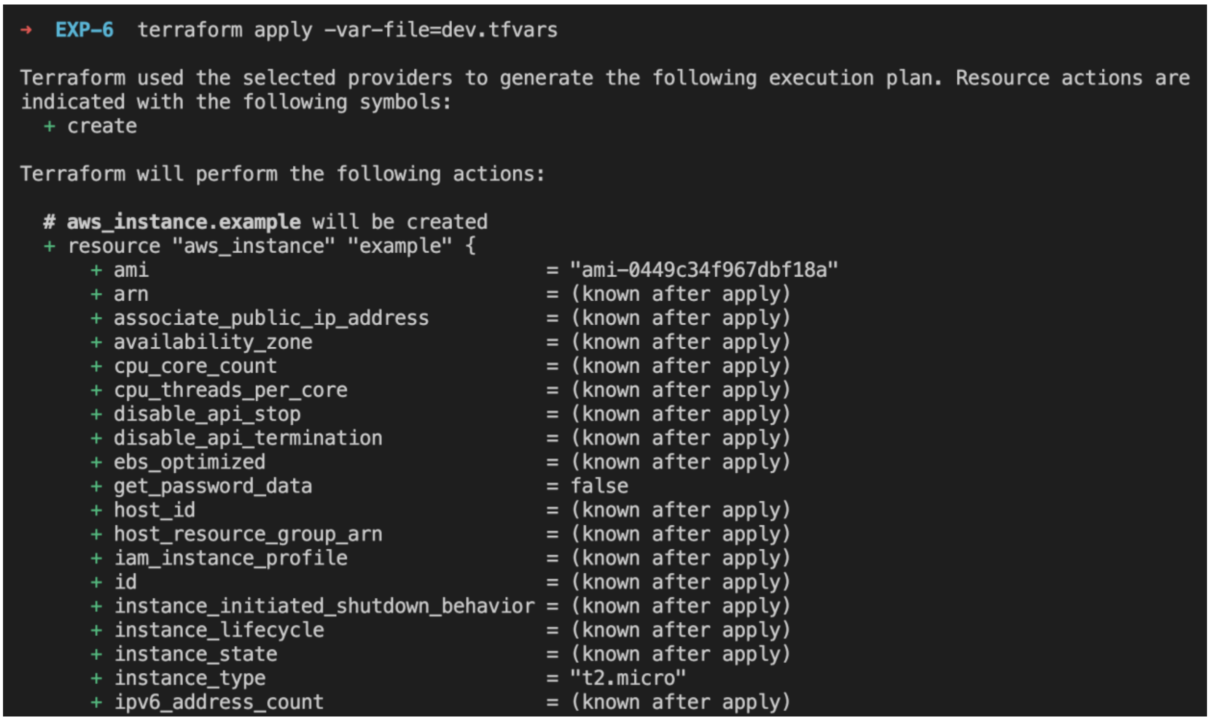


```
region        = "ap-south-1"
ami           = "ami-0449c34f967dbf18a"
instance_type = "t2.small"

```

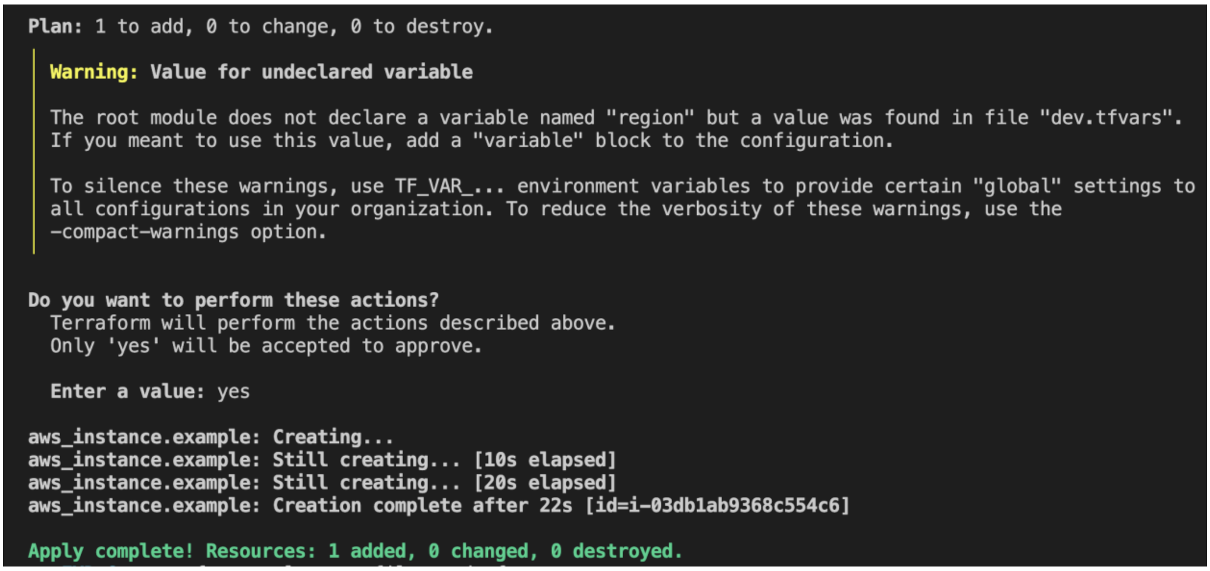
- In these files, provide values for the variables based on the environments.

3. Initialize and Apply for Dev Environment:

- Run the following Terraform commands to initialize and apply the configuration for the dev environment:



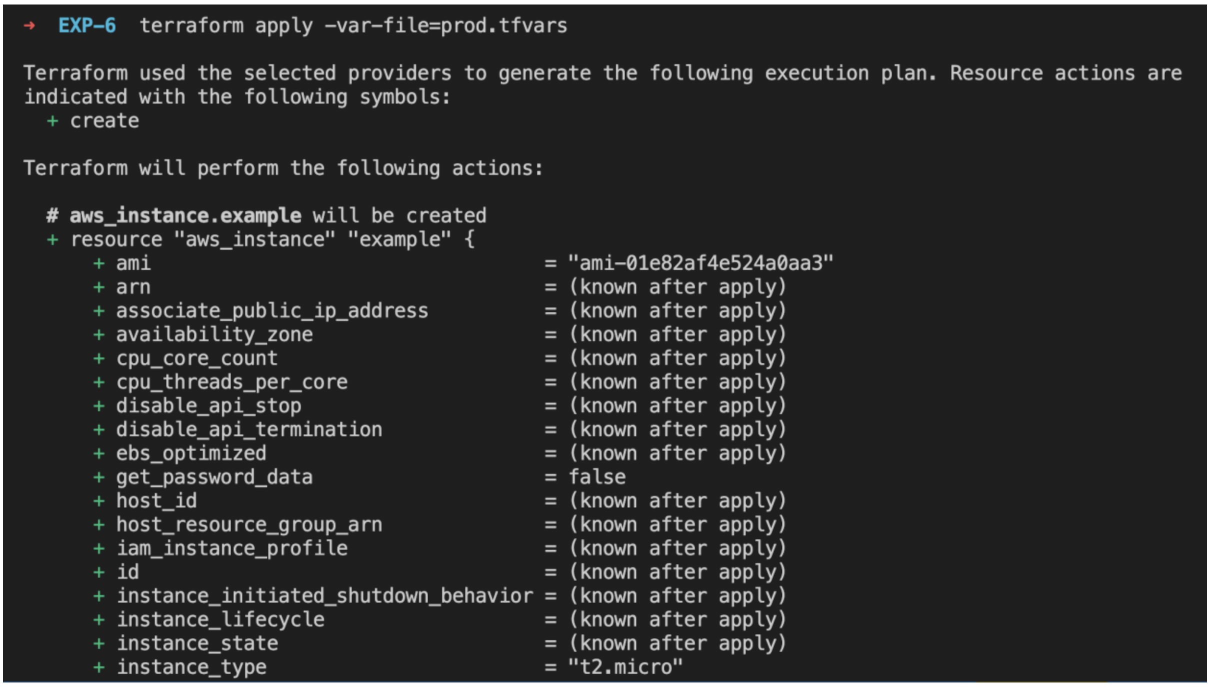
```
→  EXP-6  terraform apply -var-file=dev.tfvars

Terraform used the selected providers to generate the following execution plan. Resource actions are
indicated with the following symbols:
  + create

Terraform will perform the following actions:

  # aws_instance.example will be created
  + resource "aws_instance" "example" {
      + ami                                  = "ami-0449c34f967dbf18a"
      + arn                                  = (known after apply)
      + associate_public_ip_address          = (known after apply)
      + availability_zone                    = (known after apply)
      + cpu_core_count                       = (known after apply)
      + cpu_threads_per_core                 = (known after apply)
      + disable_api_stop                     = (known after apply)
      + disable_api_termination              = (known after apply)
      + ebs_optimized                        = (known after apply)
      + get_password_data                    = false
      + host_id                              = (known after apply)
      + host_resource_group_arn              = (known after apply)
      + iam_instance_profile                 = (known after apply)
      + id                                   = (known after apply)
      + instance_initiated_shutdown_behavior = (known after apply)
      + instance_lifecycle                   = (known after apply)
      + instance_state                       = (known after apply)
      + instance_type                        = "t2.micro"
      + ipv6_address_count                   = (known after apply)
```



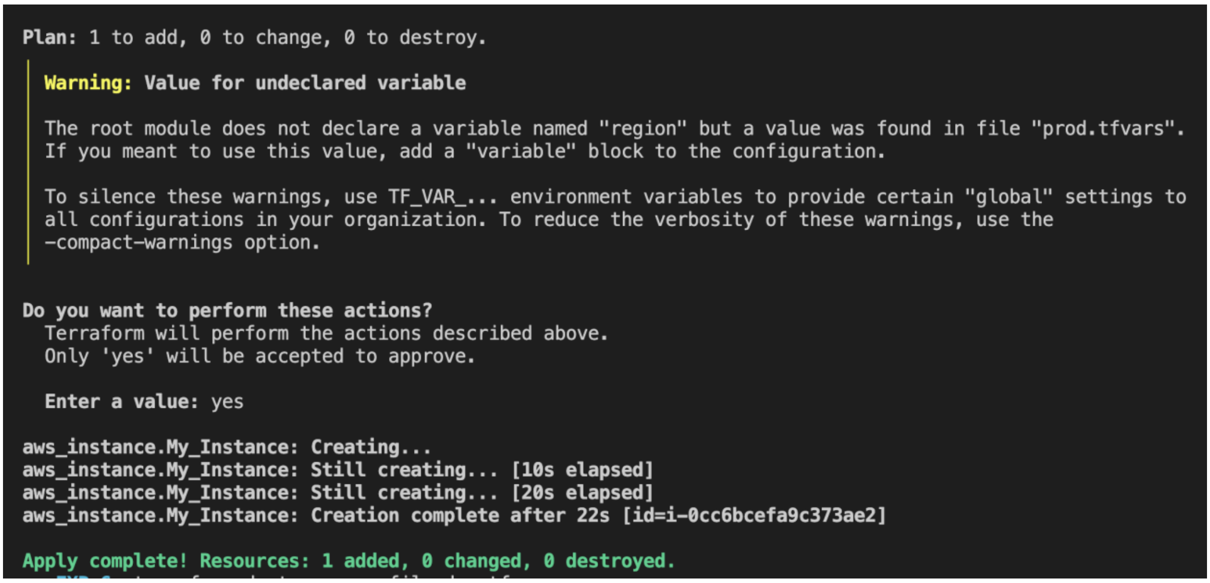
```
Plan: 1 to add, 0 to change, 0 to destroy.
│
│ Warning: Value for undeclared variable
│
│ The root module does not declare a variable named "region" but a value was found in file "dev.tfvars".
│ If you meant to use this value, add a "variable" block to the configuration.
│
│ To silence these warnings, use TF_VAR_... environment variables to provide certain "global" settings to
│ all configurations in your organization. To reduce the verbosity of these warnings, use the
│ -compact-warnings option.
│

Do you want to perform these actions?
  Terraform will perform the actions described above.
  Only 'yes' will be accepted to approve.

  Enter a value: yes

aws_instance.example: Creating...
aws_instance.example: Still creating... [10s elapsed]
aws_instance.example: Still creating... [20s elapsed]
aws_instance.example: Creation complete after 22s [id=i-03db1ab9368c554c6]

Apply complete! Resources: 1 added, 0 changed, 0 destroyed.
```

4. Initialize and Apply for Prod Environment:

- Run the following Terraform commands to initialize and apply the configuration for the prod environment:



```
→  EXP-6  terraform apply -var-file=prod.tfvars

Terraform used the selected providers to generate the following execution plan. Resource actions are
indicated with the following symbols:
  + create

Terraform will perform the following actions:

  # aws_instance.example will be created
  + resource "aws_instance" "example" {
      + ami                                  = "ami-01e82af4e524a0aa3"
      + arn                                  = (known after apply)
      + associate_public_ip_address          = (known after apply)
      + availability_zone                    = (known after apply)
      + cpu_core_count                       = (known after apply)
      + cpu_threads_per_core                 = (known after apply)
      + disable_api_stop                     = (known after apply)
      + disable_api_termination              = (known after apply)
      + ebs_optimized                        = (known after apply)
      + get_password_data                    = false
      + host_id                              = (known after apply)
      + host_resource_group_arn              = (known after apply)
      + iam_instance_profile                 = (known after apply)
      + id                                   = (known after apply)
      + instance_initiated_shutdown_behavior = (known after apply)
      + instance_lifecycle                   = (known after apply)
      + instance_state                       = (known after apply)
      + instance_type                        = "t2.micro"
```

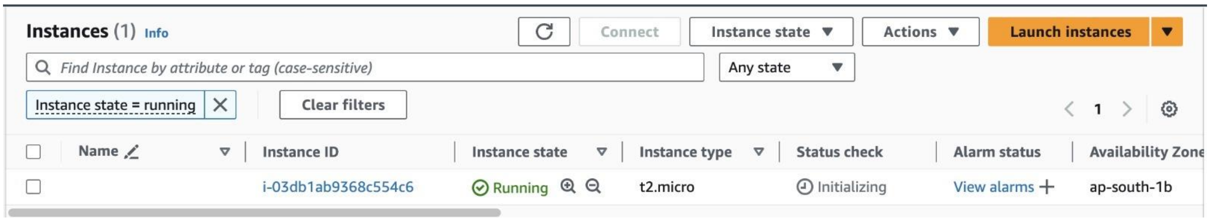


```
Plan: 1 to add, 0 to change, 0 to destroy.
│
│ Warning: Value for undeclared variable
│
│ The root module does not declare a variable named "region" but a value was found in file "prod.tfvars".
│ If you meant to use this value, add a "variable" block to the configuration.
│
│ To silence these warnings, use TF_VAR_... environment variables to provide certain "global" settings to
│ all configurations in your organization. To reduce the verbosity of these warnings, use the
│ -compact-warnings option.
╵

Do you want to perform these actions?
  Terraform will perform the actions described above.
  Only 'yes' will be accepted to approve.

  Enter a value: yes

aws_instance.My_Instance: Creating...
aws_instance.My_Instance: Still creating... [10s elapsed]
aws_instance.My_Instance: Still creating... [20s elapsed]
aws_instance.My_Instance: Creation complete after 22s [id=i-0cc6bcefa9c373ae2]

Apply complete! Resources: 1 added, 0 changed, 0 destroyed.
```

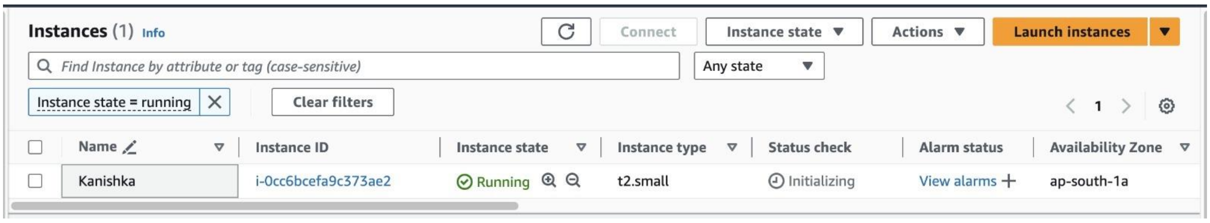
5. Test and Verify:

• Observe how different tfvars files are used to set variable values for different environments during the apply process.

• Access the AWS Management Console or use the AWS CLI to verify the creation of resources in the specified regions and instance types.



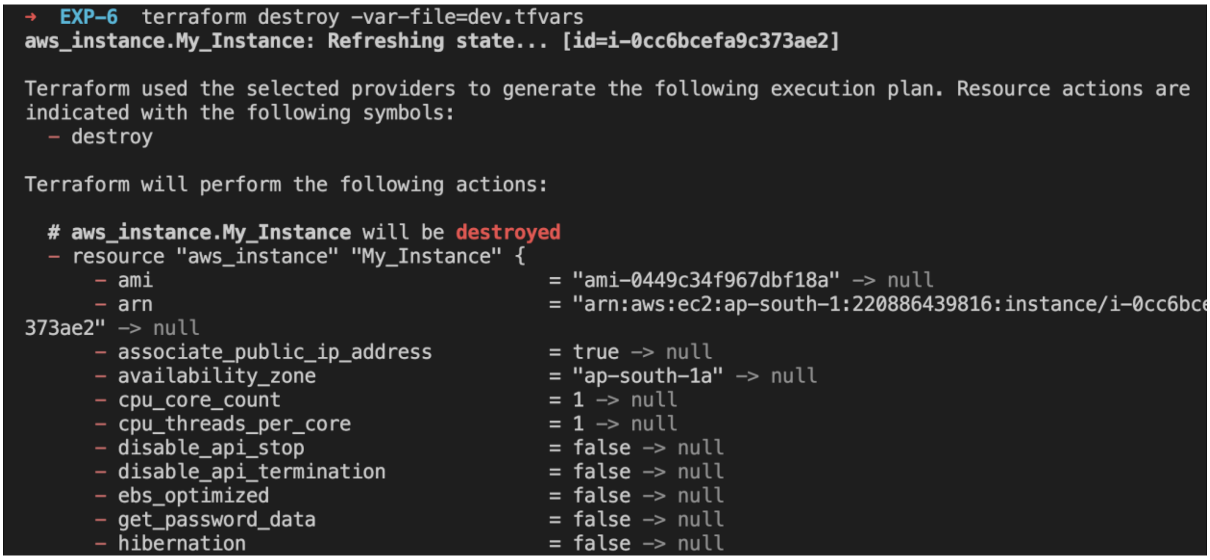
□	Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone
□		i-03db1ab9368c554c6	✓ Running	t2.micro	Initializing	View alarms +	ap-south-1b



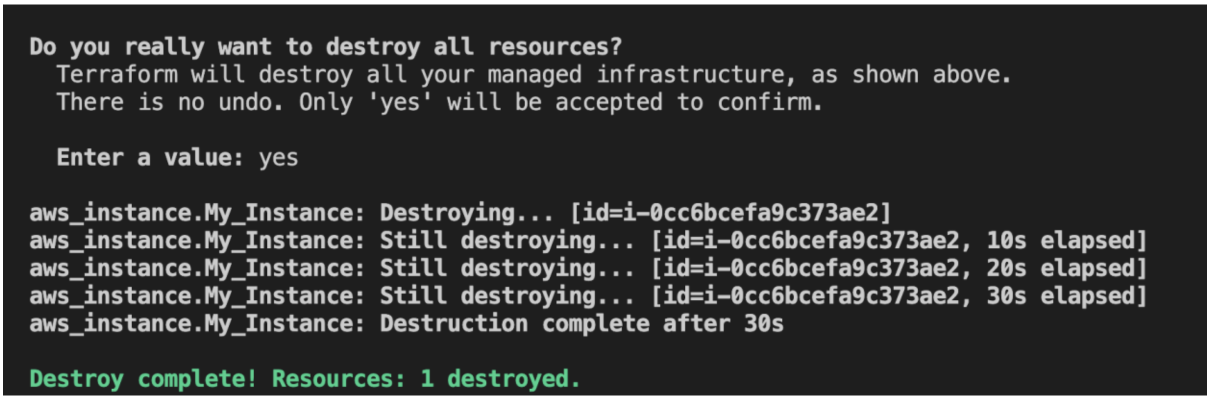
□	Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone
□	Kanishka	i-0cc6bcefa9c373ae2	Running	t2.small	Initializing	View alarms +	ap-south-1a

6. Clean Up:

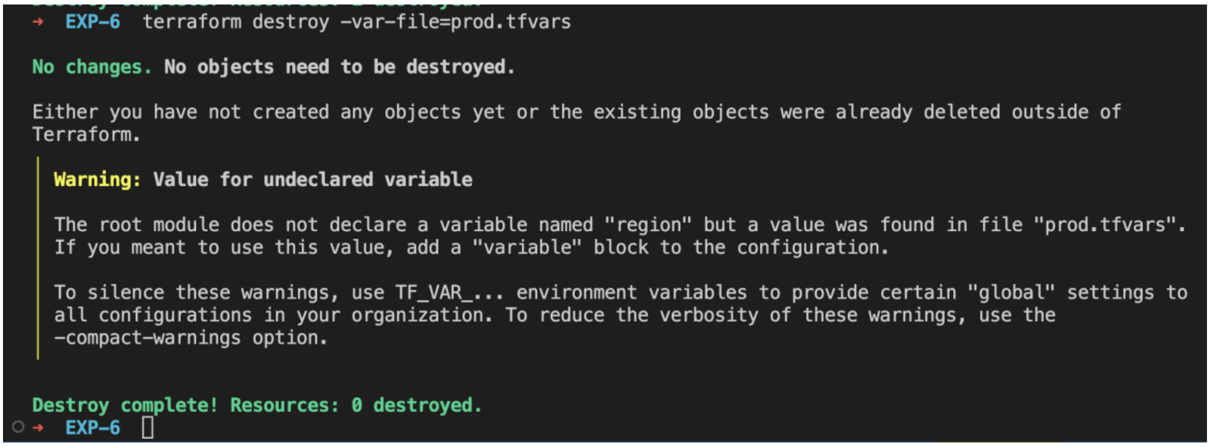
- After testing, you can clean up resources:



```
➜  EXP-6  terraform destroy -var-file=dev.tfvars
aws_instance.My_Instance: Refreshing state... [id=i-0cc6bcefa9c373ae2]

Terraform used the selected providers to generate the following execution plan. Resource actions are
indicated with the following symbols:
  - destroy

Terraform will perform the following actions:

  # aws_instance.My_Instance will be destroyed
  - resource "aws_instance" "My_Instance" {
      - ami                                  = "ami-0449c34f967dbf18a" -> null
      - arn                                  = "arn:aws:ec2:ap-south-1:220886439816:instance/i-0cc6bce
373ae2" -> null
      - associate_public_ip_address          = true -> null
      - availability_zone                    = "ap-south-1a" -> null
      - cpu_core_count                       = 1 -> null
      - cpu_threads_per_core                 = 1 -> null
      - disable_api_stop                     = false -> null
      - disable_api_termination              = false -> null
      - ebs_optimized                        = false -> null
      - get_password_data                    = false -> null
      - hibernation                          = false -> null
```



```
Do you really want to destroy all resources?
  Terraform will destroy all your managed infrastructure, as shown above.
  There is no undo. Only 'yes' will be accepted to confirm.

  Enter a value: yes

aws_instance.My_Instance: Destroying... [id=i-0cc6bcefa9c373ae2]
aws_instance.My_Instance: Still destroying... [id=i-0cc6bcefa9c373ae2, 10s elapsed]
aws_instance.My_Instance: Still destroying... [id=i-0cc6bcefa9c373ae2, 20s elapsed]
aws_instance.My_Instance: Still destroying... [id=i-0cc6bcefa9c373ae2, 30s elapsed]
aws_instance.My_Instance: Destruction complete after 30s

Destroy complete! Resources: 1 destroyed.
```



```
➜  EXP-6  terraform destroy -var-file=prod.tfvars

No changes. No objects need to be destroyed.

Either you have not created any objects yet or the existing objects were already deleted outside of
Terraform.
│
│ Warning: Value for undeclared variable
│
│ The root module does not declare a variable named "region" but a value was found in file "prod.tfvars".
│ If you meant to use this value, add a "variable" block to the configuration.
│
│ To silence these warnings, use TF_VAR_... environment variables to provide certain "global" settings to
│ all configurations in your organization. To reduce the verbosity of these warnings, use the
│ -compact-warnings option.
│

Destroy complete! Resources: 0 destroyed.
○ ➜  EXP-6
```